

学生→教員→事務局 [様式第4号の2]
平成30年月日

非接触で操作可能なデバイスの製作

システム科学技術学部 機械知能システム学科
2年 高橋 滉太
指導教員 システム科学技術学部 機械知能システム学科
教授 佐藤 和人
准教授 間所 洋和
学生支援スタッフ システム科学技術学部 機械知能システム学科
4年 松井 悠馬

1. 背景・目的

聴覚障がい者のコミュニケーションツールの1つとして、手話が存在する。しかし、聴覚障がい者と意思疎通を図るために、私たち一人ひとりが手話を習得するには時間がかかり過ぎる。そこで私は効率性に着目して、指や手の動きといったジェスチャーを通した非接触で操作できる機器を製作したいと考えた。これを応用すれば、リアルタイムでの手話翻訳機の製作も可能である。本自主研究では、指や手の動き（以後「ジェスチャー」とする。）を用いた画像解析、拡張現実などの情報付加技術について理解を深めることを目的とする。

2. 開発環境

本研究では、Windows 8(64bit)の OS 上に Visual Studio Community 2017 の IDE を導入し、C/C++ 言語で開発を行った。また、画像処理のために、拡張モジュールである `opencv_contrib` をコンパイルし、OpenCV 3.2.0 を導入した。AR の技術を使用するために ARToolKit を導入した。自作の 3D オブジェクトを作成するためにオープンソースの Metasequoia4 を使用した。

3. 提案手法

3.1 アルゴリズム

図1にアルゴリズムのフローチャートを示す。カメラを起動してから、画像を取得する。（カメラが起動しない場合、終了する。）取得した画像をグレースケールで保存したのち、任意の閾値を設定しそれより値が大きければ白、小さければ黒に変換する二値化処理により白と黒のみの画像（以下「二値化画像」とする。）を生成する。本実験では、背景や影で不鮮明な画像が取得されることを懸念し、閾値を自動で算出する自動二値化で実験を進めていく。図2に画像取得から二値化までの流れを示す。

画像判別では2枚の画像の相関係数を比較する手法と、行ごとの白と黒の変わり目を求めるパターン解析の手法で比較する。なお画像判別手法の詳細については、3.3節で明記する。画像判別で求めることができた値から、それぞれ用意したARのプログラムを実行する。

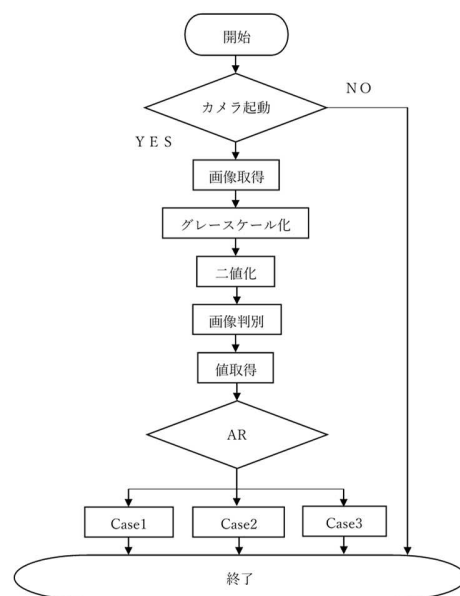


図1 アルゴリズムのフローチャート



図2 画像取得から二値化処理までの流れ

3.2 データセット

図3に画像判別に用いた入力画像を示す。二値化画像同士の相関係数による判別において、以下の3枚の画像とカメラで取得した画像を比較する。データセットとして二値化画像を選択したのは、本プログラム上での無駄な記述を省き、高速化を図ったためである。

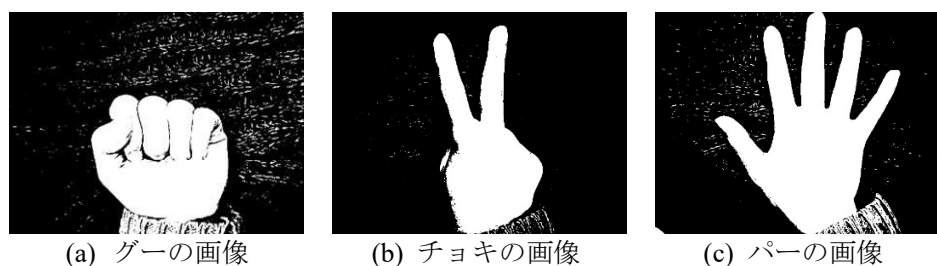


図3 画像判別に用いた入力画像

3.3 画像判別手法

1) 二値化画像同士の相関係数による判別

事前に用意しておいた3枚の画像とカメラから取得した画像をそれぞれグレースケールで保存した後、二値化処理によりピクセルの成分が白と黒のみの画像を作成する。それぞれの画像において、1ピクセルごとに色が同じかそうでないかの判別を繰り返し相関係数を求め、2枚の画像類似度を判別する。試行回数10回のうち、何回正しく判別するか試験し、認識率を算出する。認識率の計算式を式(1)に示す。(認識率を r 、試行回数を n 、誤認識数を e とする。)

$$r = \frac{(n - e)}{n} \times 100 \quad (1)$$

2) パターン解析

カメラから取得した画像をグレースケールに変換した後、二値化処理によりピクセルの成分が白と黒のみの画像を作成する。1行の中でピクセルが白から黒へ、または黒から白へ変わった回数を各行で算出した後、最大の値から判別する。これにより指の数が算出できる。理論上、グーの場合は1回、チョキの場合は4回、パーの場合少なくとも8回以上カウントされる。試行回数10回のうち、何回正しく判別できたかを試験する。また上と同様に認識率を算出する。

4. 実験結果

表1に判別結果を示す。

表1 判別結果 [%]

		出力					
		相関係数			パターン解析		
		ゲー	チョキ	パー	ゲー	チョキ	パー
入力	ゲー	40	30	30	30	40	30
	チョキ	20	40	40	0	70	30
	パー	20	30	50	0	20	80

相関係数の結果は、ゲー、チョキ、パーのいずれも誤認識が多かった。パターン解析の結果は、チョキとパーに関しては概ね良い値を得ることができた。ゲーに関しては誤認識が他と比べ多く見られた。

5. ARによるオブジェクト生成

ARToolKitのsampleファイルの内容の書き換えてプログラムを実装した。1枚の画像から3パターンの判別を行うので、少なくとも3種類のプログラムを作成する。プログラムの構成としては、カメラから画像を取得し続けるメインループ関数と3Dオブジェクトの描画を行う描画関数、キーボードの入力で終了できるような終了処理関数を作成する。また本実験ではマーカー上に3Dオブジェクトをオーバーレイするが、3Dオブジェクトのファイルの拡張子が「.mqo」であったため、GLMetaseqというライブラリを導入した。また、ヘッダファイル「gl.h」はダウンロードしたままでは、複数の構文エラーが表示され、正しくコンパイルされなかった。そこで「gl.h」のプログラム内に「windows.h」をインクルードした結果、エラーが解決した。図4に使用したHMDを示す。また図5にARによるオブジェクト表示の結果を示す。



図4 使用したHMD

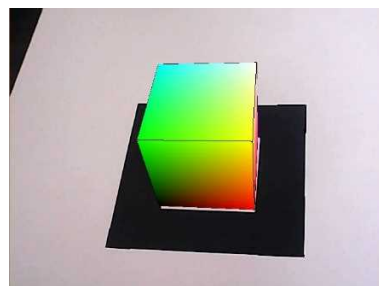


図5 ARによるオブジェクト表示

6. 考察

相関係数の判別結果については、この解析法では手のみを検出しているのではなく、背景も含めた画像全体を比較対象としていることが認識率が低かったことの原因と考えられる。これでは比較用として事前に用意していた画像との手の位置関係まで考慮しなくてはならないことが分かった。この場合、カメラと手の距離により画像に写る手の位置、大きさにばらつきが生じ、同じ形としても正しく認識しないケースが多数見られた。また、角度をつけた画像（手が斜めに写っている）なども正しく認識しなかった。対応策としては、画像のピクセル同士の関係性で判別するのではなく、形で判別できるような処理にすることが必要である。

パターン解析であれば、相関係数による判別の課題であったピクセル同士の関係性でなく形で判別できると考えた。厳密には指の本数を算出しているの形ではないが、この方法であればカメラと手の距離の問題や大人と子どもでの手の大きさの違いにも対応できると考えた。判別結果については、いくつかの誤認識が見られた。図6に誤認識時の出力画像を示す。図6の

(a)を入力画像として解析した場合、くぼみの部分に影ができ黒く判定されてしまっているため、条件上カウントされたと考えられる。また図6の(b)を入力画像として解析した場合、図6の(a)と同様にくぼみの部分の誤認識だけでなく、背景の物体も認識されてしまい上手く判別できなかったと考えられる。



図6 グーの誤認識

7. 結論

- ・2種類の方法で画像解析を試みた結果、相関係数による判別方法よりもパターン解析による判別方法の方が良い結果を得ることができた。
- ・ARToolKitを用いてディスプレイ上にオブジェクトをオーバーレイすることができた。
- ・プログラミング言語を理解し、複数のライブラリを導入することができた。

8. 課題

画像解析時の誤認識が課題となった。特にパターン解析での判別では、完全に背景と手の領域で色を分ける必要がある。カメラで取得した画像をRGBでなくHSVで保存すること、収縮処理などを用いてノイズを除去すること、背景と隔離するためにカメラと手の距離を算出することによって、認識率が向上すると考えられる。

また、形状の判別だけでなく、「右から左への移動」や「形状の変化」なども判別できると本来の目的であるジェスチャーの認識の幅が広がると考えられる。

9. 参考文献

- ・改訂第5版 ANSIC 対応 はじめてのC / 著者：楠田実 / 2015.4.10
- ・改訂版 デジタル画像処理の基礎と応用 / 著者：酒井幸市 / 2016.2.1
- ・OpenCVによるコンピュータビジョン・機械学習入門 / 著者：中村恭之・小枝正直・上田悦子 / 2017.8.4
- ・さらに進化した画像処理ライブラリの定番 OpenCV3 基本プログラミング / 著者：北山洋幸 / 2016.5.1
- ・ARToolKit 拡張現実感プログラミング入門 / 著者：橋本直 / 2010.9.14